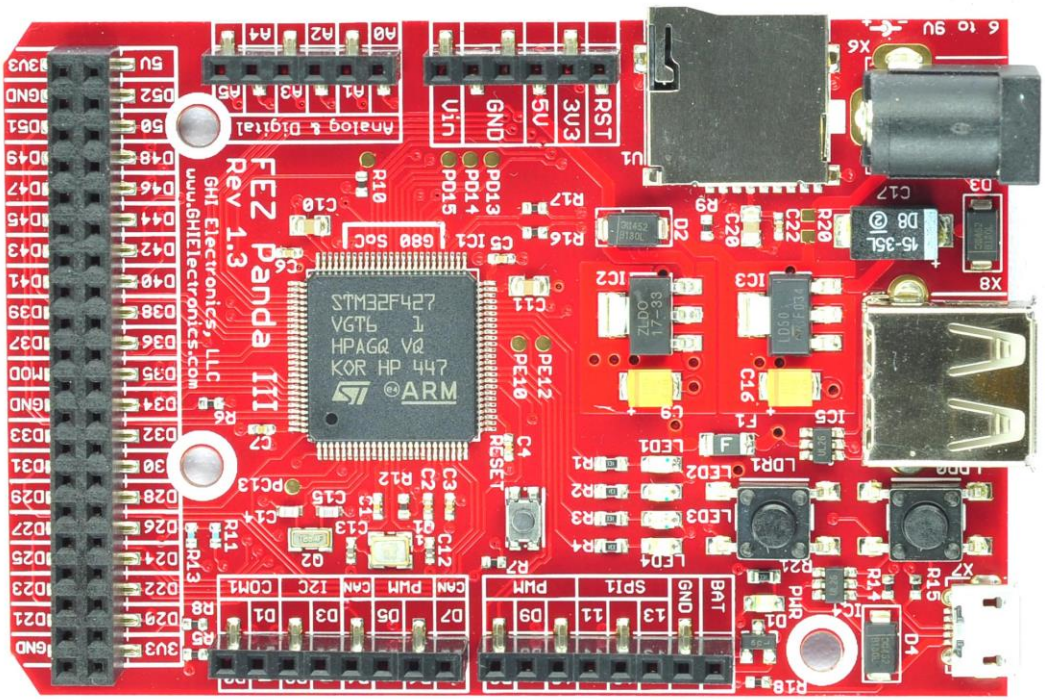
GHI Electronics
501 E. Whitcomb Ave.
Madison Heights, Michigan 48071
Phone : (248) 397-8856
Fax: (248) 397-8890
www.GHIElectronics.com

FEZ Panda III User Manual

*** DRAFT ***



Where Hardware Meets Software

FEZ Panda III User Manual ***DRAFT ***

April 29, 2015

Overview

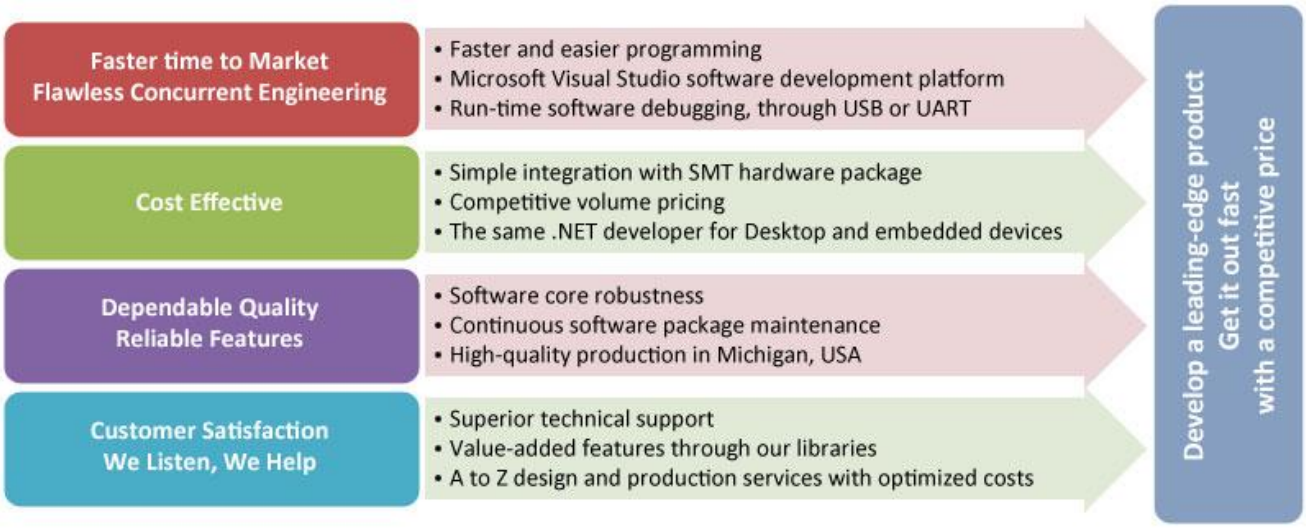
The FEZ Panda III is a small low-cost Single Board Computer (SBC) that has the G80 System on Chip (SoC) at its heart. It runs .NET Micro Framework software platform; a tiny version of Microsoft .NET framework. The value of FEZ Panda III is not only in the hardware capabilities such as the Cortex-M4 processor, memory and peripherals, but also is in the integration between the hardware and the embedded software. This provides high level features such as FAT file system, TCP/IP stack, Graphics and Threading through .NET APIs. Furthermore, the embedded software includes GHI Electronics' extensions such as USB Host and Signal Generate. All are provided royalty-free with the FEZ Panda III SBC.

All programming and debugging features are available through the state of the art Microsoft's Visual Studio. Allowing C# and Visual Basic development over a simple USB or serial connection.

FEZ Panda III User Manual ***DRAFT ***

April 29, 2015

Benefits


Faster time to Market
Flawless Concurrent Engineering
• Faster and easier programming
• Microsoft Visual Studio software development platform
• Run-time software debugging, through USB or UART
Cost Effective
• Simple integration with SMT hardware package
• Competitive volume pricing
• The same .NET developer for Desktop and embedded devices
Dependable Quality
Reliable Features
• Software core robustness
• Continuous software package maintenance
• High-quality production in Michigan, USA
Customer Satisfaction
We Listen, We Help
• Superior technical support
• Value-added features through our libraries
• A to Z design and production services with optimized costs
Develop a leading-edge product
Get it out fast
with a competitive price

Key Features

- G80 SoC at its core
- 256KByte of RAM
- 1MByte of Flash
- USB Host/Device with drivers
- 4-bit SD card interface
- Plenty of essential peripherals such as GPIO, SPI, UART, I2C, CAN, ADC, DAC and PWM.
- High level features such as file system, networking and Graphics.
- Supports Visual C# and Visual Basic

Applications

- Data Logger
- Hand Held Testers
- Internet of Things Applications
- Networked Alarm Systems
- Automation Applications
- Controllers, Robotics

FEZ Panda III User Manual ***DRAFT ***

April 29, 2015

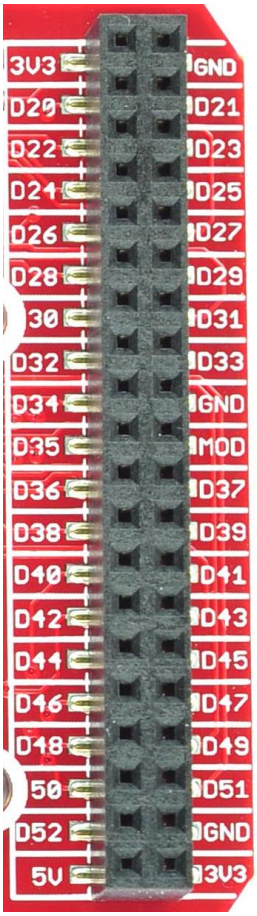
Specifications

Dimensions W x L x H (mm)	26.67 x 38.1 x TBA
Core System Hardware	G80 SoC
Processor	180 Mhz 32-bit Cortex-M4
FLASH Available/Free	1MB/256kB
RAM Available/Free	256kB/152kB
Color TFT Display Controller	Through Native SPI
Graphics (font/controls)	Complete
Image Decoder	BMP
Native Networking Support	None
Programmable IOs	52 IOs + 4 LEDs + 2 Buttons
PWM	21 + 4 LEDs
Analog Input	16
Analog Output	2
UART (COM)	4
SPI	2
I2C	Available
CAN	2
One-wire	Supported on all IOs
USB Host	HID, Mass Storage, CDC, Raw
USB Client	HID, Mass Storage, CDC, Raw
4bit SDHC/SD/MMC	Supported
Real Time Clock	Available
Piracy Protection	Available
In-Field Update	No
Operating Temperature	-40° to +85°
Lead Free	Yes
RoHS Compliant	Yes
Load native C/assembly	TBD

April 29, 2015

Pinout – 40 Pin Header

Secondary Features	Pin
3.3 Volts Out	3V3
Digital In / Out	D20
Digital In / Out	D22
Digital In / Out	D24
Digital In / Out	D26
Digital In / Out	D28
CAN2 RD	D30
CAN2 TD	D32
Digital In / Out	D34
SPI2 SCK / PWM	D35
SPI2 MISO / Analog In	D36
SPI2 MOSI / Analog In	D38
COM3 RX	D40
COM3 TX	D42
COM3 CTS	D44
COM3 RTS / PWM	D46
PWM	D48
PWM	D50
Digital In / Out	D52
5 Volts In / Out	5V



Secondary Features	Pin
Ground	GND
PWM	D21
Analog In	D23
Analog In	D25
Analog In	D27
Analog In	D29
COM4 RX / Analog In	D31
COM4 TX / Analog In	D33
Ground	GND
Mode	MOD
COM2 RX	D37
COM2 TX	D39
COM2 CTS	D41
COM2 RTS	D43
Digital In / Out	D45
Digital In / Out	D47
Digital In / Out	D49
Digital In / Out	D51
Ground	GND
3.3 Volts Out	3V3

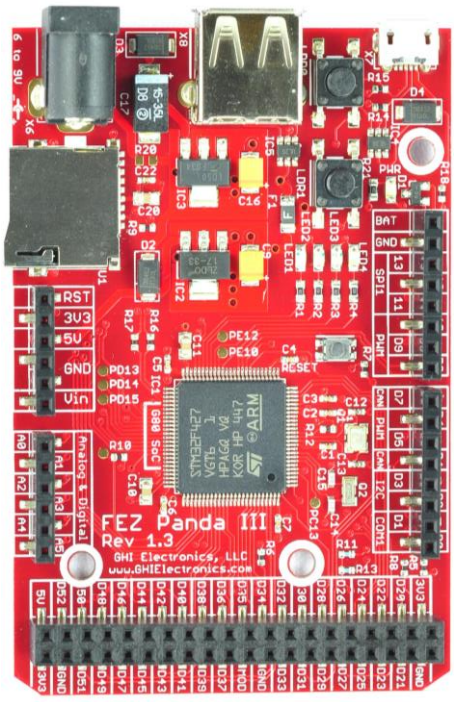
FEZ Panda III User Manual ***DRAFT ***

April 29, 2015

Pinout – Arduino-Compatible Headers

Secondary Features	Pin
Reset Input	RST
3.3 Volts Out	3V3
5 Volts In / Out	5V
Ground	GND
VIN for external	VIN

Analog In / PWM	A0
Analog In / PWM	A1
Analog In / Out	A2
Analog In / Out	A3
Analog In / PWM	A4
Analog In / PWM	A5



Secondary Features	Pin
Optional battery for RTC	VBAT
Ground	GND
SPI1 SCK / PWM	D13
SPI1 MISO / PWM	D12
SPI1 MOSI / PWM	D11
PWM	D10
Analog In / PWM	D9
Analog In / PWM	D8
CAN1 TD	D7
PWM	D6
PWM	D5
CAN1 RD	D4
I2C SCL	D3
I2C SDA	D2
COM1 TX	D1
COM1 RX	D0

April 29, 2015

FEZ Panda III and Arduino Comparison

FEZ Panda III is not an Arduino board, but it makes use of the idea of having a stackable hardware platform. The similar form factor between FEZ Panda II and Arduino allows developers to use almost any of the available Arduino shields. While using the same shields, FEZ Panda III offers more powerful hardware and software platform, greater flexibility and far more features. Starting with Microsoft Visual C# Express and the possibility for debugging and ending with high-end libraries like USB device, threading, XML, better Ethernet networking and many others. Additionally FEZ Panda III has extra IOs exposed on an easily accessible 40-pin female header.

Software Examples

These are quick snippets for a quick start up. Full details are located in the free NETMF for Beginners book, located at https://www.ghielectronics.com/support

Blinking an LED

```
using System;
using System.Threading;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using GHI.Pins;

public class Program
{
   public static void Main()
   {
      var LED = new OutputPort(FEZPandaIII.GpioLed1,true);
      while(true){
         LED.Write(!LED.Read()); // Invert
         Thread.Sleep(30);
      }
   }
}
```

Fading an LED

```
using System;
using System.Threading;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using GHI.Pins;

public class Program
{
   public static void Main()
   {
      var LED = new PWM(FEZPandaIII.PwmOutput.LED1, 10000, 0.1,
        false);
      double level = 0.5;
      double step = 0.01;
      while (true)
      {
         LED.DutyCycle = level;
         LED.Start();

         level += step;
         if ((level >= 0.9) || (level <= 0.1))
         {
            step *= -1; // Invert the step
         }
         Thread.Sleep(10);
      }
   }
}
```

Press the Button to set the LED

```
using System;
using System.Threading;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using GHI.Pins;

public class Program
{
   static OutputPort LED = new OutputPort(FEZPandaIII.Led1, false);

   public static void Main()
   {
      var Button = new InterruptPort(FEZPandaIII.Ldr0, true,
      Port.ResistorMode.PullUp, Port.InterruptMode.InterruptEdgeBoth);
      Button.OnInterrupt += Button_OnInterrupt;
      Thread.Sleep(Timeout.Infinite);
   }

   static void Button_OnInterrupt(uint pin, uint state, DateTime time)
   {
```

```
      LED.Write(state == 0);
   }
}
```

Reading a USB Keyboard

```
using System;
using System.Threading;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using GHI.Pins;
using GHI.Usb.Host;

public class Program
{
   public static void Main()
   {
      Controller.KeyboardConnected += Controller_KeyboardConnected;
      Controller.Start();
      Thread.Sleep(Timeout.Infinite);
   }

   static void Controller_KeyboardConnected(object sender, Keyboard e)
   {
      Debug.Print("Keyboard connected.");
      e.CharDown += e_CharDown;
   }

   static void e_CharDown(Keyboard sender, Keyboard.KeyboardEventArgs args)
   {
      Debug.Print("You Pressed: " + args.ASCII);
   }
}
```

Writing a file to a USB memory drive

```
using System;
using System.Threading;
using Microsoft.SPOT;
using Microsoft.SPOT.Hardware;
using Microsoft.SPOT.IO;
using System.IO;
using System.Text;

using GHI.Pins;
using GHI.Usb.Host;

public class Program
{
```

```
   private static AutoResetEvent evt = new AutoResetEvent(false);
   static MassStorage massStorage = null;

   public static void Main()
   {
      Controller.MassStorageConnected += Controller_MassStorageConnected;
      RemovableMedia.Insert += RemovableMedia_Insert;
      Controller.Start();
      evt.WaitOne(); // Wait for it be inserted
      massStorage.Mount();
      evt.WaitOne(); // Wait fro it to get mounted

      using (var fs = new FileStream("\\USB\\Hello.txt",
         FileMode.OpenOrCreate))
      fs.Write(Encoding.UTF8.GetBytes("Hello, World!"), 0, 13);
      massStorage.Unmount();
      Thread.Sleep(Timeout.Infinite);
   }

   static void RemovableMedia_Insert(object sender, MediaEventArgs e)
   {
      Debug.Print("Mounted.");
      evt.Set();
   }

   private static void Controller_MassStorageConnected(object sender,
     MassStorage ms)
   {
      Debug.Print("Inserted.");
      massStorage = ms;
      evt.Set();
   }
}
```

Reading files from an SD card

```
using System;
using System.Threading;
using Microsoft.SPOT;
using Microsoft.SPOT.IO;
using System.IO;
using GHI.IO;
using GHI.IO.Storage;

public class Program
{
   private static AutoResetEvent evt = new AutoResetEvent(false);

   public static void Main()
   {
      // Insert an SD card before running this code.
      SDCard SD = new SDCard();
```

```
      SD.Mount();
      bool fs_ready = false;
      RemovableMedia.Insert += (a, b) =>
      {
         fs_ready = true;
      };

      while (!fs_ready)
      {
         Thread.Sleep(50);
      }

      if (VolumeInfo.GetVolumes()[0].IsFormatted)
      {
         string rootDirectory = VolumeInfo.GetVolumes()[0].RootDirectory;
         string[] files = Directory.GetFiles(rootDirectory);
         Debug.Print("Files available on " + rootDirectory + ":");
         for (int i = 0; i < files.Length; i++)
            Debug.Print(files[i]);
      }
      else
      {
         Debug.Print("Storage is not formatted.");
      }
      // Unmount when done
      SD.Unmount();
   }
}
```

What's next?

The free .NET for Beginners book is the best place to get started. This book and others, along with many helpful tutorials and documents are located on the support section on the GHI Electronics website
http://www.ghielectronics.com/support

Last but not least, the GHI Electronics community is a wealth of information and support. The engineers are GHI Electronics monitor and contribute to the community forums around the clock. The home of the most active .NET Micro Framework community on the web http://www.ghielectronics.com/community